

功能：用德布尔算法求kk次平面B样条曲线上参数为u点处的r阶导矢(px, py)。

输入参数：kk-次数；u-参数；r-导矢阶数：

(m\_xAVertex, m\_yAVertex)-控制顶点， m\_aNode-节点矢量为受保护成员； m\_KnotMark-节点值导矢选择码，公

有成员。

输出参数：曲线上参数为u点处的r阶导矢(px, py)，若r=0，表示曲线上点p(u)；

当u为内节点值, r>0时, m\_KnotMark=0与1分别输出右导矢与左导矢。

```
void GetBPr(int r, int k, double u, double &px, double &py)
{
    CArray<double, double> tempx, tempy;
    if(r>k) {px=0; py=0; return;}
    int i, j;
    for(i=m_aNode.GetSize()-k-2; i>=0; i--)
    {
        if(u>=m_aNode[i]) break;
    }
    if(u<m_aNode[k]) i=k;
    if((k==1&&r==1)&&(u!=m_aNode[i] || (u==m_aNode[i]&&m_KnotMark==0)))
    {
        px=(m_xAVertex[i]-m_xAVertex[i-1])/(m_aNode[i+1]-m_aNode[i]);
        py=(m_yAVertex[i]-m_yAVertex[i-1])/(m_aNode[i+1]-m_aNode[i]);
    }
    if(k==1&&r==1&&m_KnotMark==1)
    {
        px=(m_xAVertex[i-1]-m_xAVertex[i-2])/(m_aNode[i]-m_aNode[i-1]);
        py=(m_yAVertex[i-1]-m_yAVertex[i-2])/(m_aNode[i]-m_aNode[i-1]);
    }
    if(i>k&&i<=m_xAVertex.GetSize()-1) //限于计算内节点的重复杂度Multiple
    {
        int Multiple=1;
        int ii=i;
        while(u==m_aNode[ii-1])
        {
            Multiple++;
            ii--;
        }
        if(k>1&&m_KnotMark==1&&u==m_aNode[i]&&i!=k) i=i-Multiple;
    }
    if((k==1&&r==0) || k>1)
    {
        for(j=i-k; j<=i; j++)
        {
            tempx.Add(m_xAVertex[j]);
            tempy.Add(m_yAVertex[j]);
            double x=m_xAVertex[j], y=m_yAVertex[j];
        }
        int l=0;
        if(k>1)
        for(int l=1; l<=r; l++) //教材(7.7)式第一式的r级递推
        {
            for(j=i-k; j<=i-l; j++)
            {
                double beta=(k-l+1)/(m_aNode[j+k+1]-m_aNode[j+1]);
                int jj=j-(i-k);
                tempx[jj]=beta*(tempx[jj+1]-tempx[jj]);
                tempy[jj]=beta*(tempy[jj+1]-tempy[jj]);
            }
        }
        //德布尔算法
        for(l=1; l<=k-r; l++)
        {
            for(j=i-k; j<=i-l-r; j++)
            {
                double alpha, du=m_aNode[j+k+1]-m_aNode[j+r+1];
                if(du==0.0) alpha=0.0;
                else alpha=(u-m_aNode[j+r+1])/du;
                tempx[j-i+k]=(1-alpha)*tempx[j-i+k]+alpha*tempx[j-i+k+1];
                tempy[j-i+k]=(1-alpha)*tempy[j-i+k]+alpha*tempy[j-i+k+1];
            }
        }
        px=tempx[0];
        py=tempy[0];
        tempx.RemoveAll();
        tempy.RemoveAll();
    }
}
```

}

GetBPr. txt